CONF-921125--5

Solution of Large, Sparse Systems of Linear Equations in Massively Parallel Applications

Mark T. Jones Paul E. Plassmann

Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

ANL/CP--77516

DE93 004243

Abstract

We present a general-purpose parallel iterative solver for large, sparse systems of linear equations. This solver is used in two applications, a piezoelectric crystal vibration problem and a superconductor model, that could be solved only on the largest available massively parallel machine. Results obtained on the Intel DELTA show computational rates of up to 3.25 gigaflops for these applications.

1 Overview

The computational kernel of many large-scale applications [9, 10], and in particular the two applications we discuss in this paper, is the repeated solution of large, sparse linear systems. The focus of our paper is the scalable solution of such sparse systems. As part of two separate efforts, we have implemented two massively parallel scientific applications: computation of the vortex configurations of type-II superconductors, and modeling of vibrational modes of piezoelectric crystal strip oscillators. The solution of sparse linear systems is the dominant computation of each of these applications and also the most difficult portion of each application to parallelize. We have designed and implemented scalable parallel algorithms for solving sparse linear systems that allow each of these applications to efficiently utilize scalable architectures such as the Intel DELTA.

The sparse linear algebra algorithms and software that we have designed are general purpose; they can be used for both structured and unstructured problems. The software can solve symmetric sparse problems with an arbitrary sparsity structure. The basic iterative method we use is the conjugate gradient algorithm preconditioned by an incomplete matrix factorization. A new, parallel multicoloring heuristic is employed to reorder the linear systems to obtain scalable performance [11]. Our algorithms have demonstrated scalable performance over a wide range of problems on the Intel iPSC/860 and the Intel DELTA [7]. In addition, we have achieved computational rates of up to 3.25 gigaflops on the Intel DELTA computer for sparse problems arising from the two applications considered.

We emphasize three important aspects of the performance results reported in this paper. First, the sparse iterative methods we have employed are the best *general-purpose serial* algorithms for these problems. Through the development of new techniques, we have made these previously unparallelizable serial algorithms scalable. Higher computational rates could be achieved by using an inferior iterative method at the expense of much greater total solution time. Second, these scalable methods are independent of any specific matrix structure; higher computational rates could be achieved by using an implementation that is sparsity structure specific for each application. Third, we note that only *effective* flops are included in the computational rates we have reported, namely, only those flops that would take place in a good serial implementation.

Software portability has been a major consideration in the design of our scalable linear algebra libraries. We encapsulate the communication in a few subroutines and utilize macros to interface to a number of different message-passing packages. We use the Level 2 and 3 *dense* BLAS in the computationally intensive portions of the code. We are able to make use of the higher-level BLAS by extracting common nonzero structures and local clique structures from these sparse systems using graph techniques. These techniques are described briefly in §2. The high-level dense BLAS en-

able us to achieve excellent performance on the high-performance RISC chips that are utilized in most massively parallel machines.

We demonstrate the general applicability of our software through its use in two very different applications. The matrix sparsity structures, condition numbers, and eigenvalue distributions of the linear systems that arise in these two applications are radically different.

The first application involves the computation of the equilibrium vortex structure and configurations for three-dimensional, layered type-II superconductors. The physical configuration of these systems are modeled by computing the minimum energy solution to a generalization of the Ginzburg-Landau free energy given by the Lawrence-Doniach model. By using large-scale optimization techniques, we can successfully compute solutions to these models for complex vortex structures. Because solutions are desired for arbitrary applied magnetic fields and the material is nonisotropic, a complete three-dimensional model is required. The resulting problems are very computationally demanding, often requiring the solution of nonlinear systems with 10^7 independent variables. In this collaborative project with the Materials Science Division at Argonne, the computed solutions are being used to develop a more complete understanding of vortex behavior and, therefore, enable the development of higher-temperature superconductors. Sustained computational rates of over 3 gigaflops for the entire application have been achieved on the Intel DELTA. For one problem instance, we have found improvements of over a *factor of 100* in the total execution time when compared to the same problem run on the CRAY-2. In addition, the ability to run significantly larger problems on the Intel DELTA allows for the solution of much more accurate three-dimensional superconductor models.

The second application, a collaborative project between Argonne and Motorola, involves the modeling of the vibrational modes of piezoelectric crystal strip oscillators. Piezoelectric crystals are crucial to the performance of almost every product Motorola manufactures; they are the critical component of almost all oscillatory circuits. These crystals must be designed to vibrate in a particular mode shape at a specific frequency over a wide range of temperatures. To meet these demanding design goals, engineers would like to be able to accurately model the behavior of the crystals in a timely fashion from their desktop workstation. A special high-order finite element formulation developed by engineers at Motorola and Argonne is used to model the electrical and mechanical characteristics of the crystals. Because we are interested in a particular subset of modes near the middle of the eigenspectrum, we must use a very refined mesh, making this a very computationally demanding problem. Accurate modeling of these vibrational modes requires the solution of sparse matrices with more than 10^8 nonzeros. By utilizing the Intel DELTA at a sustained computation rate of approximately 2 gigaflops, we have solved problems in a little over a hour that engineers at Motorola had previously abandoned as computationally intractable.

Both the sparse linear algebra codes and the application codes operate in a non trivial environment. The codes themselves are complex; the linear algebra software constitutes over 10,000 lines, as does each of the application codes. They must operate in a portable fashion; we routinely run the codes on networks of workstations, the BBN TC2000, the Intel iPSC/860, and the Intel DELTA. Each of the codes interfaces to a number of graphics packages such as AVS or the Doré system. In addition, the crystal code is capable of sending data from the parallel machine that it running on a socket connection to a graphics workstation for instantaneous display.

In §2 we discuss the scalable sparse linear algebra libraries. In §3 and §4 we describe the two applications, the superconductor vortex configuration problem and the vibrational mode problem for piezoelectric strip oscillators. Finally, in §5 we present the performance results for these two applications on the Intel DELTA.

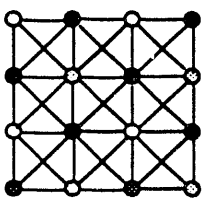
2 Scalable Iterative Methods for the Solution of Sparse Linear Systems

In this section we briefly discuss the algorithms used in the scalable libraries for solving sparse linear systems. The iterative solver uses an incomplete matrix factorization as a preconditioner for the conjugate gradient algorithm [12]. This is a general-purpose preconditioner that performs well for many problems, including the sparse problems arising in the applications with which we are concerned. In fact, examples of the linear systems that arise in these problems cannot be efficiently solved with a simple local preconditioner. An important advantage of this approach is that it can be used for structured and unstructured problems.

The traditional serial approach to generate incomplete factors employs a "natural" ordering of the variables. Unfortunately, a scalable parallel implemen-

tation of this approach is impossible because the dependencies in the solution of triangular systems make this computation inherently sequential. However, a reordering of the preconditioning matrix based on a graph coloring does allow for its scalable solution. The reordered triangular system solution is scalable because the number of sequential communication steps is proportional to the chromatic number of the graph, which is essentially a function of the local graph structure and independent of the size of the graph. In Figure 1 we give an example of a multicoloring ordering for a regular grid that requires four colors.

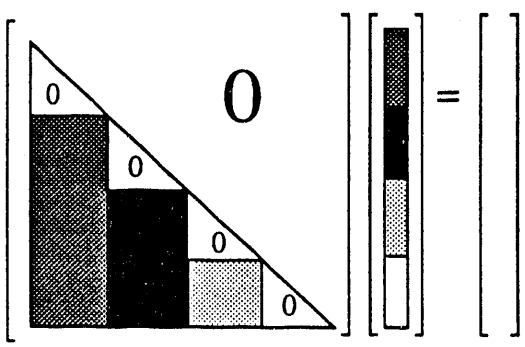
Figure 1: The adjacency graph corresponding to a nine-point stencil requires four colors. An ordering of the variables of the corresponding linear system allows a triangular system of the same structure to be solved in four major parallel steps. One step for the unknowns corresponding to each color, followed by interprocessor communication to update the right-hand side.



9-pt Stencil

4 Colors

Matrix Reordered by Color



We have demonstrated that this technique is effective for unstructured problems [7]. These algorithms have exhibited scalable performance for a range of structured and unstructured finite element and finite difference problems. We have also developed and implemented a scalable graph coloring heuristic based on finding a sequence of independent sets [11]. Thus, this approach is not dependent on the *a priori* knowledge of a coloring. In addition, we note that recent theoretical results have shown that one does not see the dramatic increase the number of iterations required for convergence with "many color" orderings that one sees with the red/black ordering for model problems [8]. These results corroborate our experimental convergence results for the sparse problems for these two applications.

It is not sufficient to achieve scalable performance; one must also achieve good computation rates on each node. For RISC chips such as the i860, the best performance is obtained by algorithms that exhibit good data locality and minimize indirect addressing. A standard implementation of a sparse matrix times vector multiplication does not exhibit good data locality and uses a large amount of indirect addressing. To improve locality and minimize indirect addressing, one can take advantage of the special local structure inherent to many of these multicomponent problems. For example, large, dense cliques exist in these graphs and can be easily recognized. In all of the sparse operations that we perform, operations involving these cliques can utilize dense level 2 and 3 BLAS. In addition, many rows of the sparse matrix have identical structure, but differing nonzero values. By exploiting this structure, we can significantly reduce the amount of indirect addressing. We note that these ideas have been used with dramatic effect in direct sparse factorization for several years. In §5 we demonstrate the improvement in processor efficiencies obtained for these systems.

3 Computation of Equilibrium Vortex Configurations for 3-Dimensional Layered Type-II Superconductors

The recent discovery of new, high-temperature superconductors has resulted in a tremendous interest in the modeling and understanding of layered type-II superconducting systems. Characteristic of type-II superconductors is their ability to remain superconducting in a so-called mixed state. In the mixed state, which exists between a lower and upper critical applied magnetic field denoted by H_{c1} and H_{c2}, respectively, these materials allow magnetic flux lines to penetrate the bulk of the material. This phenomenon is possible because of the formation of compensating vortex currents around these magnetic flux lines which shield the remaining superconducting regions from the effect

of the magnetic field. This physical property of type-II superconductors is very different from the original type-I variety, which are superconducting only at very low temperatures and are characterized by the expulsion of any applied magnetic field from the bulk of the material (the well-known Meissner effect).

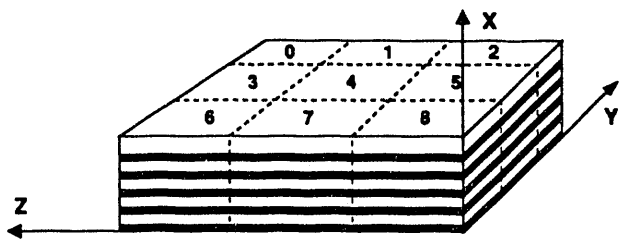
An understanding of and predictive capability for the vortex structure and configurations of vortices in type-II superconductors are crucial to the development of desired physical properties for these materials. As a phenomenological model, the Ginzburg-Landau theory has been very successful in the prediction of the vortex structure and configurations for type-II superconductors. Unfortunately, analytic solutions to this model are known only for a few special cases. Therefore, the development of effective computational methods for solving this model are essential to its use in a predictive capacity.

In this section we briefly introduce the problem formulation and discuss a number of numerical optimization algorithms that have been attempted. The most successful of these algorithms is a damped, inexact Newton method. The main computational task in the Newton method is the approximate solution of a sparse linear system. We also discuss the program complexity and the I/O and graphics requirements of the implementation. In §5 we present experimental results obtained on the Intel DELTA for several representative problems.

3.1 Problem Formulation

The Ginzburg-Landau (GL) model is based on the observation that the local free energy of a superconductor is adequately represented by the first few terms of its functional expansion in terms of a (complex-valued) order parameter. When the effects of an applied magnetic field are included in the model, one obtains an expression for the free energy that depends on the order parameter, its spatial gradient, and a set of vector potentials that describe the magnetic field within the superconducting material. Although the effects of mass anisotropy can be accounted for in the Ginzburg-Landau model, it is often not suitable for modeling materials with alternating insulating-superconducting layers, in which the order parameter may vary discontinuously in the direction normal to the layers. An extension of the GL model by Lawrence and Doniach (LD) does allow for variation of this type in the order parameter, and hence is more appropriate for layered materials. The LD model is used to obtain the results presented in this paper.

Figure 2: The three-dimensional layered model partitioned in two dimensions



The free-energy functional is defined on a three-dimensional rectangular mesh with the geometric layout depicted in Figure 2. Shown are the alternating layers of superconducting and insulating material considered by the Lawrence-Doniach model. When the model is discretized, a finer degree of resolution is generally given to the insulating layers. The complex order parameter is defined within the superconducting layers, and the vector potential is defined within the insulating layers. The number of magnetic flux quanta and the angle of the applied magnetic field are enforced by the imposition of quasi-periodic boundary conditions on the model. For the problems of interest, the number of grid points necessary to represent the model in the direction perpendicular to the layers (the X axis in Figure 2) is smaller than the number of points required in the two directions parallel to the layers (the Y axis and Z axis in Figure 2). We make use of this property and partition the grid in the Y and Z directions. For example, in Figure 2 the Y-Z domain is shown partitioned among nine processors.

We obtain numerical solutions for the order parameter and vector potential by applying finite-difference discretization techniques to the free-energy expression for the LD model and then using large-scale optimization techniques to solve the resulting discrete problem. The solution of this problem yields an approximation to the equilibrium superconducting electron distribution that actually minimizes the free energy.

The order parameter and vector potential data required to evaluate the function, gradient, and Hessian on the local grid assigned to a processor includes both the local grid points and nonlocal, adjacent grid points. In our implementation, the interprocessor communication required to obtain these nonlocal values is managed by the BlockComm communication package, developed by William Gropp [5].

3.2 Optimization Techniques

The original attempts at determining a minimizer of the Ginzburg-Landau free-energy functional were done by Doria et al. [2] for a homogeneous (non-layered) two-dimensional model. They used a Monte-Carlo simulated annealing algorithm and were able to obtain solutions only for small grid sizes with at most two vortices per unit cell. These calculations required millions of iterations and many hours of Cray time; eventually they were forced to resort to using the simpler London model to try to make the problem tractable. The simulated annealing method is most suitable for problems in which many local minima (i.e., metastable configurations) are present and for which derivatives of the function are difficult to obtain. When these conditions do not hold, local optimization methods are more appropriate.

A second group, Wang and Hu, made this observation and used a first-order technique: a steepest descent algorithm with fixed step size [14]. With this method they were able to obtain solutions, for the homogeneous model, for which the gradient norm was reduced to no less than 10^{-5} on two-dimensional grids of size up to 200×200.

More sophisticated techniques were introduced by Garner et al. [4] with greatly improved results. Among these methods were three classes of optimization methods: nonlinear conjugate gradient, limited-memory quasi-Newton, and modified Newton methods. Even though they require only the same kind of derivative information as the steepest descent method (that is, first derivatives), these algorithms have much better convergence properties. Typically, the first-order methods, limited-memory quasi-Newton, and nonlinear conjugate gradients require several thousand iterations to reduced the gradient norm to 10^{-7}. Unfortunately, a good local minimizer is not always obtained by the first-order methods. The inexact Newton method, a second-order method developed by Plassmann and Wright, is much more robust. This method usually requires fewer than 20 iterations and can reduce the gradient norm to less than 10^{-12}. However, this method requires the approximate solution of a large, sparse linear system at each iteration.

Thus, the only two viable optimization methods are an efficient first-order method (we have found the limited memory BFGS, or L-BFGS, method to be the most effective) and the inexact, damped Newton method. For the present problem, the storage requirement for the inexact Newton method is comparable to that of L-BFGS and depends on problem configuration. However, each iteration of the damped Newton method requires much more work than an iteration of L-BFGS, but of course many fewer iterations are required (typically, the iteration count is less than 25, as opposed to many thousands for L-BFGS).

3.3 Complexity of the Implementation

The complete implementation for the superconductivity application is a very large and complex program. More than 13,000 lines of Fortran and C are involved, not including the sparse linear algebra code or the BlockComm code used for the interprocessor communication required for the function and gradient evaluation. The code to analytically evaluate the Hessian involves more than 6,000 lines of Fortran.

On the Intel DELTA parallel I/O to the concurrent file system (CFS) is used to collect order parameter and vector potential data from the processors for graphical analysis. Each processor is responsible for writing out its partition of the data set to the appropriate place on the CFS. Currently, both AVS and XDataSlice (from NCSA) are used to analyze the three-dimensional data.

4 Modeling Vibrational Modes of Piezoelectric Crystals

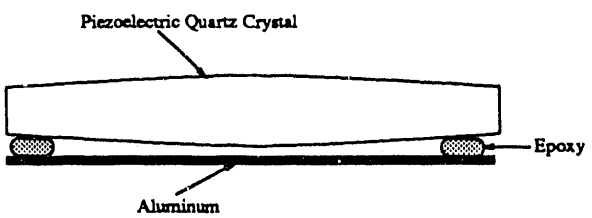
Piezoelectric crystals are an important component in electronic appliances such as computers, cellular phones, and pagers. To be useful, these crystals must resonate in a particular vibrational mode at a specified frequency over a wide range of temperatures. Two major barriers prevent the accurate and timely modeling of these crystals. First, existing finite-element software packages (e.g., NASTRAN) are unable to accurately model the physical properties of piezoelectric systems. Second, the computational requirements of these problems are so large that engineers have considered their solution to be intractable on current sequential computers.

Extensive collaboration between Argonne National Laboratory and Motorola has resulted in the development of algorithms and software to overcome these two barriers. We have developed a new finite-element formulation of this problem that is capable of accurately modeling crystal behavior. To satisfy the computational requirements of the problem, we have developed scalable algorithms and software to harness the cost-efficient power of the Intel DELTA.

4.1 Problem Formulation

The piezoelectric crystals that are currently the focus of our modeling effort are "strip" oscillators. These crystals are thin strips of quartz that vibrate at a fixed frequency when an electric current is applied to the crystal. A diagram of a strip oscillator affixed to an aluminum substrate with epoxy is shown in Figure 3.

Figure 3: Strip oscillator



The quartz crystal is anisotropic; that is, the crystal is cut at an angle that is not necessarily parallel to any of its axes. The vibration frequency of the crystal that we are primarily interested in is the fundamental thickness-shear mode. In this mode, the primary displacement is along the longest axis of the crystal and is concentrated in the center of the crystal. The fundamental thickness-shear mode is near the middle of the eigenspectrum of the crystal, making it difficult to isolate. There may be several eigenmodes near the fundamental thickness-shear mode. Because these modes may interfere with the operation of the crystal, we are also interested in them.

We use the finite element method to model the crystal. An efficient 27-node brick element was developed at Argonne and Motorola to accurately model these anisotropic crystals [1]. A second-order Lagrangian polynomial is used to interpolate the mechanical and electric field potential; second-order interpolation is necessary to accurately model the thickness-shear mode. A large displacement formulation is used to include nonlinear geometric effects that result from a combination of thermal loads and displacement constraints. Because we are interested in eigenmodes in the middle of the eigenspectrum, we must use a large number of elements to accurately model the behavior of the crystal.

The solution phase has two steps. First, we find the deformation of the crystal due to thermal displacement. For example, if the crystal was mounted on aluminum at 25C, it will deform when the temperature is increased to 35C. This requires solving a nonlinear static thermal stress problem. Second, we find the vibrational modes of interest for the deformed crystal. This requires solving a linear vibration problem; a generalized eigenproblem.

To solve the nonlinear static thermal stress problem, we must solve a series of linear systems of the form $Ku = f$, where K represents the stiffness matrix, u represents the displacements, and f represents the forces resulting from thermal loads and displacement constraints. The major task here, of course, is the solution of very large sparse systems of equations.

To solve the linear vibration problem, we must solve a generalized eigenproblem of the form $Kx = \omega^2 Mx$, where K represents the stiffness matrix, M represents the mass matrix, x is a vibrational mode shape, and ω is a vibrational mode. We use a shifted, inverted variant of the Lanczos algorithm to solve this eigenproblem [13]. This method has been shown to be very efficient for the parallel solution of the vibration problem [6]. Again, the major computational task is the solution of large sparse systems of linear equations.

The key to accurately modeling these crystals is our new, second-order finite element. The key to timely solution is the scalable solution of large, sparse systems of linear equations.

4.2 Parallel Implementation

The three-dimensional finite element grid needed to model the crystals is much more refined in the length and width directions than it is in the thickness direction. We can take advantage of this fact and partition the grid among the processors in only the length and width directions. This approach reduces communication and maps nicely onto the DELTA architecture. Each processor is assigned a rectangular solid corresponding to a portion of the grid. Each processor is responsible for evaluating the finite elements in its partition and for maintaining all relevant geometric and solution data for that partition. The code for these activities is over 8,000 lines. All aspects of this code have been implemented in a scalable fashion.

We collect geometric information, thermal displacements, and vibrational modes and mode shapes on the CFS, the concurrent file system on the Intel DELTA and Intel iPSC/860. Each processor is responsible for writing out its portion of data corresponding to the *surface* of the crystal to the appropriate place on the CFS. The data is written in a form that can be post-processed by either the portable AVS graphics system

or the Dorè graphics system on the Stardent Titan machine. The graphics systems are used to display thermal displacements, and electric field potentials and to animate vibrational mode shapes.

5 Performance on the Intel DELTA

First, we present experimental results for the superconductivity problem obtained on the Intel DELTA for three representative problem sets. In Tables 1 and 2 we give the relevant parameters for these problems. NX, NY, and NZ are the number of grid points used in the X, Y, and Z directions, respectively. NK is the discretization used within a layer for the vector potential; thus, the number of layers is given by NX/NK. The number of vortices per unit cell induced by the quasi-periodic boundary is given by VORNUM. The number of independent variables is given by N, and the number of nonzeros in the Hessian is given by NNZ.

Timings for the L-BFGS method and the damped, inexact Newton method described in §3 are given in Tables 1 and 2. Shown are the number of iterations required, the norm of the gradient at termination, and the sustained computational rate for the inexact Newton method. For comparison, we have included the L-BFGS running time for PROBLEM-3 on a CRAY-2.

PROBLEM-1 is representative of a problem where the resolution of the vector potential within the insulating layers is important. This resolution is required when the vortex coherence length is small relative to the interlayer spacing. For problems where coarser resolution of the layers can be allowed, as in PROBLEM-2 and PROBLEM-3, larger numbers of layers (NX/NK) can be investigated. PROBLEM-2 and PROBLEM-3 differ in their geometric shape, the angle of the applied magnetic field, and the number of vortices used.

The difficulty of the problem solution greatly increases as the number of vortices and the problem size is increased. For the problems in Tables 1 and 2, L-BFGS was at best able to obtain approximate solutions; for all these problems L-BFGS was not able to determine a solution that satisfied the convergence criteria, and thus was not always physically meaningful. For the results next to the symbol (**) the line search was unable to satisfy the Wolfe conditions, and for the results next the symbol (*) the maximum number of L-BFGS iterations was exceeded. In general, L-BFGS is unable to adequately solve problems for which VORNUM is greater than two, whereas the inexact Newton code has been able to obtain solutions for hundreds of vortices per unit cell. Thus, to obtain accurate solutions large problems or for problems involving larger numbers of vortices, the inexact Newton method must be used. Finally, in comparison with the performance of L-BFGS on the CRAY-2, we note that an improvement of approximately 95 times was obtained on the Intel DELTA for PROBLEM-3 [3], or an improvement of more than a 100 times when compared to the inexact Newton method.

Table 1: Problem set description

	PROBLEM-1	PROBLEM-2
NX	24	64
NK	8	4
NY	80	64
NZ	96	96
VORNUM	4	4
N	6.0×10^5	1.6×10^6
NNZ	2.0×10^8	1.7×10^8
L-BFGS (DELTA)	45.5 min. (**)	61.3 min. (*)
L-BFGS ($\|\|g\|\|$)	1.2×10^{-8}(**)	5.0×10^{-7} (*)
L-BFGS (iters.)	14,664 (**)	10,000 (*)
Newton (DELTA)	39.4 min.	43.2 min.
Newton ($\|\|g\|\|$)	3.0×10^{-12}	1.0×10^{-11}
Newton (iters.)	22	27
Newton (Gflops)	3.25	2.55

We emphasize that all the symbolic computation required to reorder the linear systems is done in parallel. The parallel coloring heuristic used is described in [11], and the clique reduction algorithm is described in [10]. Note that the coloring is not done for the original graph but, rather, for the quotient graph obtained by modding out by the identical node structure and the local cliques. For example, in PROBLEM-3 the original graph size is 1.8×10^6, and the size of the clique graph is 7.3×10^4. Thus, the size of the graph colored (the clique graph) is actually quite modest. The time required to determine the clique graph on the DELTA for PROBLEM-3 is 3.359 seconds, and the time required to color the clique graph is 0.059 seconds. These operations are done only once, since the matrix structure does not change between iterations; also these times are negligible with respect to the total solution times. Other characteristics of the clique graph for PROBLEM-3 are that the average degree of the graph is 105 and the number of colors required by the parallel heuristic is 11.

Table 2: Problem set description (continued)

	PROBLEM-3
NX	20
NK	2
NY	150
NZ	150
VORNUM	1
N	1.8×10^6
NNZ	1.9×10^8
L-BFGS (CRAY-2)	$\approx$ 40 hr.
L-BFGS (DELTA)	25.3 min. (*)
L-BFGS ($\|\|g\|\|$)	1.0×10^{-6} (*)
L-BFGS (iters.)	3,100 (*)
Newton (DELTA)	22.1 min.
Newton ($\|\|g\|\|$)	1.0×10^{-12}
Newton (iters.)	17
Newton (Gflops)	1.38

In Table 3 we note the effect of varying the layer discretization on the i860 processor performance during the solution of the linear systems. For these numbers we have used 128 processors and fixed the local problem size to be roughly equivalent. The second column shows the number of identical nodes found in the graph by the solver; the third column shows the average clique size found. The final column shows the average computational rate per processor during the solution of the linear systems.

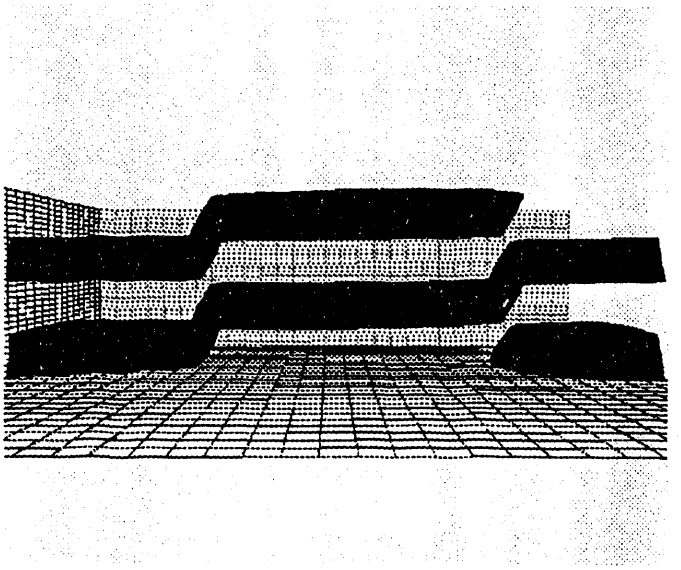
Table 3: Effect of varying the layer discretization on the processor performance in solving the linear systems

NK	I-node size	Avg. clique size	Avg. Mflops/ processor
2	8	32.0	2.97
4	14	44.8	5.42
6	20	60.0	6.71
8	26	78.0	8.96

The model has been used to extract detailed information about the three-dimensional vortex structure. The advent of massively parallel computers and the development of efficient computational algorithms have made this first "computational" glimpse into a layered, type-II superconductor possible.

For example, in Figure 4 we show isosurfaces of the magnetic field magnitude for a system near H_{c2}. This solution was obtained by using the inexact, damped Newton method. The external magnetic field is applied at an angle fixed by the pseudo-periodic boundary conditions imposed on the model. This view is taken parallel to the layers of the material; we can see the vortices tilted with respect to the layers. One can observe the "pinching" of the vortices as the magnetic field isosurfaces pass through the superconducting sheets and then spread out again in the insulating layers. In addition, one can see the characteristic "staircase" structure of the vortices that has been observed experimentally when the applied magnetic field is tilted.

Figure 4: View parallel to the layers of the vortex structure in the layered three-dimensional model



It is energetically favorable for a vortex to remain parallel to a superconducting layer. As a consequence, there can be a difference between the vortex angle in the bulk of the material and the angle of the applied field. If the angle of the applied field is close to being parallel to the layers, a "vortex locking" phenomenon has been predicted and can be observed computationally with this model.

We now present results for the piezoelectric crystal application on the Intel DELTA. We have solved problems consisting of over 480,000 equations with 161,150,990 nonzeros on 512 processors of the Intel DELTA. Over 99 percent of the time is spent in solving the linear systems and evaluating the finite elements. The solution of the linear systems has achieved speeds of approximately 2 gigaflops on 512 processors.

This speed is scalable; the individual processor performance degrades only from 4.16 megaflops per processor to 3.83 megaflops per processor when one goes from 128 processors to 512 processors and keeps the subgrid size fixed. This performance is particularly impressive given that we are using a *general* sparse matrix solver. The evaluation of the finite elements achieves over 6 gigaflops on 512 processors, this speed is completely scalable as the evaluation can take place independently on each processor. The overall code achieves a speed of approximately 2 gigaflops on 512 processors. Thus, in little over an hour of wall-clock time, we can now solve problems that engineers at Motorola deemed intractable.

The vibrational mode of interest to engineers at Motorola is the fundamental thickness-shear mode of the crystal. This mode is characterized by vibration primarily along the long axis of the crystal and occurs at 4.2 Mhz. This frequency is significantly higher than the fundamental vibrational mode of the crystal. We wish to vary both temperature and mounting methods and observe the changes in the vibration modes near 4.2 MHz. However, it is important to identify all other nearby modes, generally those modes within a few KHz. For example, the action of a nearby extensional mode, a spurious mode characterized by opposing motions along the face of the crystal, can interfere with the performance of the crystal. It has recently become possible to compare our computational data with experimental laser interferometry data generated at Motorola. This comparison shows agreement between the surface dispacements between model and experiment for both the fundamental thickness-shear mode and nearby spurious vibrational modes.

Acknowledgments

This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. We acknowledge our scientific collaborators for the superconductivity application: James Garner (Bradley University), Roy Benedek, Shelly Shumway, and Katherine Strandburg (Materials Science Division, Argonne), and Stephen Wright and David Levine (Mathematics and Computer Science Division, Argonne). We also acknowledge our scientific collaborators for the piezoelectric crystal application: Thomas Canfield (Computing and Telecommunications Division, Argonne) and Michael Tang (Motorola, Inc.).

References

[1] T. R. CANFIELD, M. S. H. TANG, AND J. E. FOSTER, *Non-linear geometric and thermal effects in three dimensional Galerkin finite-element formulations for piezoelectric crystals*, computing and telecommunications division, Argonne National Laboratory, Argonne, IL, 1991.

[2] M. M. DORIA, J. E. GUBERNATIS, AND D. RAINER, *On solving the Ginzburg-Landau equations by simulated annealing*, Physical Review B, 41 (1990), p. 6335.

[3] J. GARNER, (*Private Communication*). Materials Science Division, Argonne National Laboratory, April 1992.

[4] J. GARNER, M. SPANBAUER, R. BENEDEK, K. STRANDBURG, S. WRIGHT, AND P. PLASSMANN, *Critical fields of Josephson-coupled superconducting multilayers*, Physical Review B, 45 (1992), pp. 7973–7983.

[5] W. GROPP, *BlockComm - A communication package for block transfers* (unpublished manuscript), Argonne National Laboratory, Mathematics and Computer Science Division, 1992.

[6] M. T. JONES AND M. L. PATRICK, *The Lanczos algorithm for the generalized symmetric eigenproblem on shared-memory architectures*, Preprint MCS-P182-1090, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1990.

[7] M. T. JONES AND P. E. PLASSMANN, *Scalable iterative solution of sparse linear systems*, Preprint MCS-P277-1191, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1991.

[8] ——, *The effect of many color orderings on the convergence of iterative methods*, in Proceedings of the Copper Mountain Conference on Iterative Methods, SIAM LA-SIG, 1992.

[9] ——, *Solution of large, sparse systems of linear equations in massively parallel applications*, Preprint MCS-P313-0692, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1992.

[10] ——, *The efficient parallel iterative solution of large sparse linear systems*, in Proceedings of the IMA Workshop on Sparse Matrix Computations: Graph Theory Issues and Algorithms, October 14–18, 1991, (to appear).

[11] ——, *A parallel graph coloring heuristic*, SIAM Journal on Scientific and Statistical Computing, (to appear).

[12] T. A. MANTEUFFEL, *An incomplete factorization technique for positive definite linear systems*, Mathematics of Computation, 34 (1980), pp. 473–497.

[13] B. NOUR-OMID, B. N. PARLETT, T. ERICSSON, AND P. S. JENSEN, *How to implement the spectral transformation*, Mathematics of Computation, 48 (1987), pp. 663–673.

[14] Z. D. WANG AND C. R. HU, *A numerical relaxation approach for solving the general Ginzburg-Landau equations for type-II superconductors*, Physical Review B, 44 (1991), p. 11918.

END

DATE FILMED

2 / 11 / 93